



(19) Europäisches Patentamt
 European Patent Office
 Office européen des brevets



(11) Publication number:

0 535 746 A2

(12)

EUROPEAN PATENT APPLICATION

(21) Application number: 92202980.6

(51) Int. Cl. 5: H04N 5/14

(22) Date of filing: 29.09.92

(30) Priority: 30.09.91 GB 9120739
 19.08.92 GB 9217742

(43) Date of publication of application:
 07.04.93 Bulletin 93/14

(84) Designated Contracting States:
 AT CH DE FR GB IT LI NL SE

(71) Applicant: PHILIPS ELECTRONICS UK LIMITED
 Philips House 1-19 Torrington Place
 London WC1E 7HD(GB)

(84) GB

(71) Applicant: N.V. Philips' Gloeilampenfabrieken
 Groenewoudseweg 1
 NL-5621 BA Eindhoven(NL)

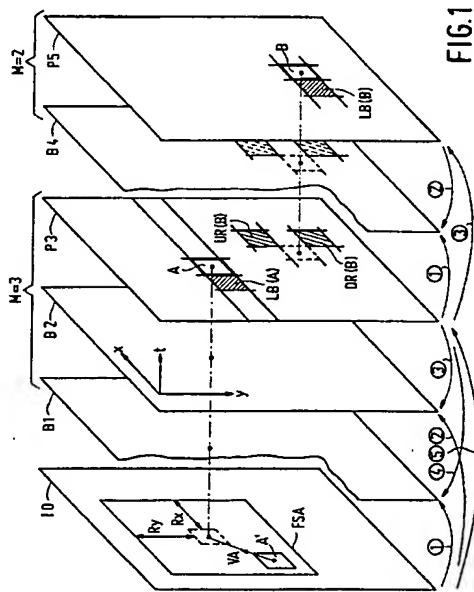
(84) CH DE FR IT LI NL SE AT

(72) Inventor: Andrew, Derek
 c/o Philips Research Laboratories, Cross
 Oak Lane
 Redhill, Surrey RH1 5HA(GB)
 Inventor: Morris, Octavius John
 c/o Philips Research Laboratories, Cross
 Oak Lane
 Redhill, Surrey RH1 5HA(GB)

(74) Representative: Gray, John James et al
 Philips Electronics Patents and Trade Marks
 Department Philips House 1-19 Torrington
 Place
 London WC1E 7HD (GB)

(54) Motion vector estimation, motion picture encoding and storage.

(57) In the encoding of a frame (P3) in a digitised motion picture sequence, the frame is divided into blocks (A) of pixels and a motion vector (VA) found to predicts the contents of each block A from a block (A') in a previous frame (I0). Rather than evaluate many thousands of candidate vectors to find the best motion vector (VA), relatively few candidate vectors are evaluated, based on starting vector which is the estimated motion vector for a neighbouring block (LB(A)) in the same frame. To allow parallel processing of several pixel blocks, all the blocks of one row in the current frame are assigned to one processor (DSP), and a starting vector is obtained from a block (LB(A)) previously processed by the same processor. Further starting vectors are obtained if desired from neighbouring blocks (UR(B)-DR(B)) in neighbouring frames of the sequence. The candidate vectors based on a starting vector may be very few in number, or may fill an area a few percent of the full search area. The method may favour candidate vectors which are more economical to encode.



EP 0 535 746 A2

The invention relates to a method of estimating motion vectors for blocks of pixels in a frame of a digitised motion picture sequence, the method comprising, for each given block of pixels performing a vector search based around at least one starting vector derived from a previously estimated vector for a pixel block which is a neighbour in space and/or time of the given pixel block.

The techniques of motion vector estimation can be used to improve compression of motion picture data by reducing the temporal redundancy in the picture information, in a manner analogous to that in which well-known differential coding and transform coding techniques eliminate spatial redundancy for the compression of still picture data.

The International Standards Organisation has issued a Committee Draft Standard ISO 11172, well-known the art as the MPEG specification and incorporated herein by reference. MPEG specifies a format for the encoding of compressed motion picture data, including motion vectors. An overview of the MPEG coding standard is provided by D. Le Gall in 'MPEG: A Video Compression Standard for Multimedia Applications', Comm. of the ACM, Vol. 34, No. 4, April 1991. MPEG encoding allows the reproduction of full-motion video from a data stream of only 1.2 Mbits per second, a rate low enough for the reproduction of video from MPEG data stored on a Compact Disc (CD-ROM), or received via some other data channel of similar bandwidth.

The MPEG specification provides for motion compensation by dividing a frame into blocks of 16 x 16 pixels, and assigning to each a motion vector in the range ± 64 pixels in the x (horizontal) and y (vertical) dimensions, to half-pixel precision. This will be explained in more detail below with reference to Figure 1 of the drawings. The MPEG specification does not specify how the motion vectors are to be obtained, only how they are to be represented. A full search for the most accurate motion vector in the range permitted would involve comparing the current pixel block with over 16000 candidate pixel blocks, even at integer pixel precision. Dedicated hardware can be provided for this task, but there is also a need for software implementations that can be used in advance of the general availability of such hardware, and which offer more flexibility to investigate improvements in the vector search and other aspects of the encoding strategy.

A method as set forth in the opening paragraph has been described by G. de Haan and H. Huijgen in their paper 'Motion estimator for TV-picture enhancement', presented at the Fourth International Workshop on HDTV, Turin, September 1991. That method has been developed for implementation in dedicated hardware and real-time operation, and

involves the evaluation of relatively few candidate blocks. The method exploits the fact that, where object features moving in the motion picture sequence are larger than a few pixels blocks, as is generally the case, there is a high degree of correlation between the motion vectors of neighbouring blocks. Thus, by careful selection of starting vectors from among vectors estimated for neighbouring blocks, and by the selection of candidate vectors predicted from the starting vectors, good quality motion estimation can be achieved with relatively few candidate block comparisons. While the known technique can be implemented in software to produce a quicker result than the full search, even using a high speed programmable digital signal processor chip (DSP) the encoding process is still inconveniently slow.

5 A possibility for increasing the speed exists in theory in the use of available hardware systems comprising several DSPs operating in parallel, each processing a subset of the blocks in each frame. Unfortunately, this speed increase is not readily realisable with the method of de Haan and Huijgen because a given DSP processing a given block 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175 180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325 330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475 480 485 490 495 500 505 510 515 520 525 530 535 540 545 550 555 560 565 570 575 580 585 590 595 600 605 610 615 620 625 630 635 640 645 650 655 660 665 670 675 680 685 690 695 700 705 710 715 720 725 730 735 740 745 750 755 760 765 770 775 780 785 790 795 800 805 810 815 820 825 830 835 840 845 850 855 860 865 870 875 880 885 890 895 900 905 910 915 920 925 930 935 940 945 950 955 960 965 970 975 980 985 990 995 1000 1005 1010 1015 1020 1025 1030 1035 1040 1045 1050 1055 1060 1065 1070 1075 1080 1085 1090 1095 1100 1105 1110 1115 1120 1125 1130 1135 1140 1145 1150 1155 1160 1165 1170 1175 1180 1185 1190 1195 1200 1205 1210 1215 1220 1225 1230 1235 1240 1245 1250 1255 1260 1265 1270 1275 1280 1285 1290 1295 1300 1305 1310 1315 1320 1325 1330 1335 1340 1345 1350 1355 1360 1365 1370 1375 1380 1385 1390 1395 1400 1405 1410 1415 1420 1425 1430 1435 1440 1445 1450 1455 1460 1465 1470 1475 1480 1485 1490 1495 1500 1505 1510 1515 1520 1525 1530 1535 1540 1545 1550 1555 1560 1565 1570 1575 1580 1585 1590 1595 1600 1605 1610 1615 1620 1625 1630 1635 1640 1645 1650 1655 1660 1665 1670 1675 1680 1685 1690 1695 1700 1705 1710 1715 1720 1725 1730 1735 1740 1745 1750 1755 1760 1765 1770 1775 1780 1785 1790 1795 1800 1805 1810 1815 1820 1825 1830 1835 1840 1845 1850 1855 1860 1865 1870 1875 1880 1885 1890 1895 1900 1905 1910 1915 1920 1925 1930 1935 1940 1945 1950 1955 1960 1965 1970 1975 1980 1985 1990 1995 2000 2005 2010 2015 2020 2025 2030 2035 2040 2045 2050 2055 2060 2065 2070 2075 2080 2085 2090 2095 2100 2105 2110 2115 2120 2125 2130 2135 2140 2145 2150 2155 2160 2165 2170 2175 2180 2185 2190 2195 2200 2205 2210 2215 2220 2225 2230 2235 2240 2245 2250 2255 2260 2265 2270 2275 2280 2285 2290 2295 2300 2305 2310 2315 2320 2325 2330 2335 2340 2345 2350 2355 2360 2365 2370 2375 2380 2385 2390 2395 2400 2405 2410 2415 2420 2425 2430 2435 2440 2445 2450 2455 2460 2465 2470 2475 2480 2485 2490 2495 2500 2505 2510 2515 2520 2525 2530 2535 2540 2545 2550 2555 2560 2565 2570 2575 2580 2585 2590 2595 2600 2605 2610 2615 2620 2625 2630 2635 2640 2645 2650 2655 2660 2665 2670 2675 2680 2685 2690 2695 2700 2705 2710 2715 2720 2725 2730 2735 2740 2745 2750 2755 2760 2765 2770 2775 2780 2785 2790 2795 2800 2805 2810 2815 2820 2825 2830 2835 2840 2845 2850 2855 2860 2865 2870 2875 2880 2885 2890 2895 2900 2905 2910 2915 2920 2925 2930 2935 2940 2945 2950 2955 2960 2965 2970 2975 2980 2985 2990 2995 3000 3005 3010 3015 3020 3025 3030 3035 3040 3045 3050 3055 3060 3065 3070 3075 3080 3085 3090 3095 3100 3105 3110 3115 3120 3125 3130 3135 3140 3145 3150 3155 3160 3165 3170 3175 3180 3185 3190 3195 3200 3205 3210 3215 3220 3225 3230 3235 3240 3245 3250 3255 3260 3265 3270 3275 3280 3285 3290 3295 3300 3305 3310 3315 3320 3325 3330 3335 3340 3345 3350 3355 3360 3365 3370 3375 3380 3385 3390 3395 3400 3405 3410 3415 3420 3425 3430 3435 3440 3445 3450 3455 3460 3465 3470 3475 3480 3485 3490 3495 3500 3505 3510 3515 3520 3525 3530 3535 3540 3545 3550 3555 3560 3565 3570 3575 3580 3585 3590 3595 3600 3605 3610 3615 3620 3625 3630 3635 3640 3645 3650 3655 3660 3665 3670 3675 3680 3685 3690 3695 3700 3705 3710 3715 3720 3725 3730 3735 3740 3745 3750 3755 3760 3765 3770 3775 3780 3785 3790 3795 3800 3805 3810 3815 3820 3825 3830 3835 3840 3845 3850 3855 3860 3865 3870 3875 3880 3885 3890 3895 3900 3905 3910 3915 3920 3925 3930 3935 3940 3945 3950 3955 3960 3965 3970 3975 3980 3985 3990 3995 4000 4005 4010 4015 4020 4025 4030 4035 4040 4045 4050 4055 4060 4065 4070 4075 4080 4085 4090 4095 4100 4105 4110 4115 4120 4125 4130 4135 4140 4145 4150 4155 4160 4165 4170 4175 4180 4185 4190 4195 4200 4205 4210 4215 4220 4225 4230 4235 4240 4245 4250 4255 4260 4265 4270 4275 4280 4285 4290 4295 4300 4305 4310 4315 4320 4325 4330 4335 4340 4345 4350 4355 4360 4365 4370 4375 4380 4385 4390 4395 4400 4405 4410 4415 4420 4425 4430 4435 4440 4445 4450 4455 4460 4465 4470 4475 4480 4485 4490 4495 4500 4505 4510 4515 4520 4525 4530 4535 4540 4545 4550 4555 4560 4565 4570 4575 4580 4585 4590 4595 4600 4605 4610 4615 4620 4625 4630 4635 4640 4645 4650 4655 4660 4665 4670 4675 4680 4685 4690 4695 4700 4705 4710 4715 4720 4725 4730 4735 4740 4745 4750 4755 4760 4765 4770 4775 4780 4785 4790 4795 4800 4805 4810 4815 4820 4825 4830 4835 4840 4845 4850 4855 4860 4865 4870 4875 4880 4885 4890 4895 4900 4905 4910 4915 4920 4925 4930 4935 4940 4945 4950 4955 4960 4965 4970 4975 4980 4985 4990 4995 5000 5005 5010 5015 5020 5025 5030 5035 5040 5045 5050 5055 5060 5065 5070 5075 5080 5085 5090 5095 5100 5105 5110 5115 5120 5125 5130 5135 5140 5145 5150 5155 5160 5165 5170 5175 5180 5185 5190 5195 5200 5205 5210 5215 5220 5225 5230 5235 5240 5245 5250 5255 5260 5265 5270 5275 5280 5285 5290 5295 5300 5305 5310 5315 5320 5325 5330 5335 5340 5345 5350 5355 5360 5365 5370 5375 5380 5385 5390 5395 5400 5405 5410 5415 5420 5425 5430 5435 5440 5445 5450 5455 5460 5465 5470 5475 5480 5485 5490 5495 5500 5505 5510 5515 5520 5525 5530 5535 5540 5545 5550 5555 5560 5565 5570 5575 5580 5585 5590 5595 5600 5605 5610 5615 5620 5625 5630 5635 5640 5645 5650 5655 5660 5665 5670 5675 5680 5685 5690 5695 5700 5705 5710 5715 5720 5725 5730 5735 5740 5745 5750 5755 5760 5765 5770 5775 5780 5785 5790 5795 5800 5805 5810 5815 5820 5825 5830 5835 5840 5845 5850 5855 5860 5865 5870 5875 5880 5885 5890 5895 5900 5905 5910 5915 5920 5925 5930 5935 5940 5945 5950 5955 5960 5965 5970 5975 5980 5985 5990 5995 6000 6005 6010 6015 6020 6025 6030 6035 6040 6045 6050 6055 6060 6065 6070 6075 6080 6085 6090 6095 6100 6105 6110 6115 6120 6125 6130 6135 6140 6145 6150 6155 6160 6165 6170 6175 6180 6185 6190 6195 6200 6205 6210 6215 6220 6225 6230 6235 6240 6245 6250 6255 6260 6265 6270 6275 6280 6285 6290 6295 6300 6305 6310 6315 6320 6325 6330 6335 6340 6345 6350 6355 6360 6365 6370 6375 6380 6385 6390 6395 6400 6405 6410 6415 6420 6425 6430 6435 6440 6445 6450 6455 6460 6465 6470 6475 6480 6485 6490 6495 6500 6505 6510 6515 6520 6525 6530 6535 6540 6545 6550 6555 6560 6565 6570 6575 6580 6585 6590 6595 6600 6605 6610 6615 6620 6625 6630 6635 6640 6645 6650 6655 6660 6665 6670 6675 6680 6685 6690 6695 6700 6705 6710 6715 6720 6725 6730 6735 6740 6745 6750 6755 6760 6765 6770 6775 6780 6785 6790 6795 6800 6805 6810 6815 6820 6825 6830 6835 6840 6845 6850 6855 6860 6865 6870 6875 6880 6885 6890 6895 6900 6905 6910 6915 6920 6925 6930 6935 6940 6945 6950 6955 6960 6965 6970 6975 6980 6985 6990 6995 7000 7005 7010 7015 7020 7025 7030 7035 7040 7045 7050 7055 7060 7065 7070 7075 7080 7085 7090 7095 7100 7105 7110 7115 7120 7125 7130 7135 7140 7145 7150 7155 7160 7165 7170 7175 7180 7185 7190 7195 7200 7205 7210 7215 7220 7225 7230 7235 7240 7245 7250 7255 7260 7265 7270 7275 7280 7285 7290 7295 7300 7305 7310 7315 7320 7325 7330 7335 7340 7345 7350 7355 7360 7365 7370 7375 7380 7385 7390 7395 7400 7405 7410 7415 7420 7425 7430 7435 7440 7445 7450 7455 7460 7465 7470 7475 7480 7485 7490 7495 7500 7505 7510 7515 7520 7525

search may be advantageous since horizontal motion is predominant in typical motion picture sequences. A set of n processors may each take every n -th row of blocks in the image frame. Alternatively, each processor may take the next unprocessed row in the frame whenever it has finished a row of pixel blocks. Many other arrangements are of course possible.

In preferred embodiments of the method, for each given pixel block at least one further prediction of the motion vector is based on a motion vector estimated for a pixel block in a neighbouring frame in the motion picture sequence. By this means starting vectors can be chosen from neighbouring blocks effectively surrounding the given block, which improves the accuracy of the estimated motion vectors when object features move over the given block from a wide range of directions.

It will be appreciated by those skilled in the art that the terms 'neighbouring' and 'preceding' as used herein with reference to frames, blocks or pixels cannot be interpreted as referring only to strictly adjacent frames, blocks or pixels. In an MPEG encoding embodiment, for example, it is clear that 'neighbouring' frames may be separated at least by a number of frame intervals of the same order of magnitude as the batch size M .

In a particular embodiment, the predictions for each given pixel block (disregarding the first block in each subset) are based on motion vectors estimated previously for:

- the said neighbouring block in the same subset as the given block; and
- two blocks in a neighbouring frame occupying spatial positions relative to the given block so as to define with the said preceding block in the same subset as the given block a triangular distribution generally surrounding the given block.

In the prediction of a motion vector on the basis of the at least one starting vector, it may be for example that matching between the given block and a block in a previous frame is measured for an array of from 5 to 10 candidate vectors distributed around the starting vector. The candidate vectors may in particular be distributed within a few pixels of the starting vector, perhaps with a horizontal bias, and may include the starting vector itself.

It will often be the case, in the known method as well as in a method embodying the invention as described above, that a good set of starting vectors is not available, and perhaps only a single default starting vector is available for the given block. Such cases arise at the edges of the frames of pixel data, and also when encoding the first predicted frame of a sequence, for example following a scene change. Although such cases are in the

minority, a full vector search may still be prohibitively time consuming.

The invention accordingly provides, in a second aspect, a method of estimating a motion vector for a given block of pixels in a frame of a digitised motion picture sequence, by performing an evaluation of a plurality of candidate vectors based around a starting vector, characterised in that the candidate vectors surround the starting vector and are evaluated in an ordered series of generally increasing distance from the starting vector. The ordered series of candidate vectors may form a spiral around the starting vector, for example. The area covered by the candidate vectors may be for example an order of magnitude smaller than that required for a full search, but, if the vector estimated for a neighbouring block is used as the starting vector, the process will tend to converge on an accurate motion vector within a few blocks, always assuming that the moving objects are larger than a few blocks.

Moreover, the ordering of the candidate vectors ensures that the best candidate vector will tend to occur sooner rather than later in the sequence, and various checks can be implemented when evaluating candidate vectors so as to terminate the evaluation of candidate vectors once the best candidate, or at least a "good enough" candidate has been found.

It is a further object of the invention to improve generally the motion picture quality encoded by a stream of motion-compensated image data.

In a third aspect, the invention provides a method of estimating a motion vector for a given block of pixels in a frame of a digitised motion picture sequence, which estimated vector is to be encoded in a data stream encoding the motion picture sequence, the method comprising:

- generating a set of candidate vectors for evaluation;
- evaluating the quality of prediction provided by each of the candidate vectors; and
- comparing the results of said evaluations to select one of the candidate vectors as the estimated vector for encoding in a stream of data encoding the motion picture sequence, wherein the amount of data required to encode the estimated motion vector is dependent on the value of the estimated motion vector and wherein the selection of the estimated vector from among the candidate vectors depends not only on the quality of prediction provided by each candidate, but also on the amount of data required to encode each candidate in a data stream, such that the candidate vector selected is not the one providing the highest quality of prediction, if it is determined that another one of the candidate vectors can be encoded in less data while providing a prediction quality which

is within a predetermined threshold of the highest prediction quality.

This third aspect of the invention enables an overall improvement in the quality of motion pictures encoded within a bit stream at a limited data rate. If the quality threshold is set appropriately, the bit stream capacity which has been freed by more economical coding of the selected motion vector can be used to improve the encoding quality of other parts of the picture sequence, even if some of the freed capacity is required for increased error information resulting from the selection of a motion vector which does not quite give the best prediction for a given block. Of course, use of the invention in its third aspect is not limited to any combination with the other aspects of the invention.

As a particular example, the MPEG coding standard provides a special format for encoding blocks which have the zero vector as their motion vector. In some circumstances a zero vector does not have to be encoded at all. In certain picture sequences, including for example cartoon animation sequences, it has been found that a significant amount of bit stream capacity is "wasted" when conventional encoders seek to encode "best" motion vectors for a large number of blocks representing, say, uniform blue sky. For such blocks, the only change from frame to frame in fact comprises only random noise superimposed upon the plain coloured area. It has been found that for these sequences the invention in its third aspect can lead to great savings in bit stream capacity, and hence great improvements in overall picture quality, when the quality threshold is set just high enough that the encoder will ignore the random noise motion information.

The invention still further provides methods and apparatuses for performing motion vector estimation, and for encoding and storing motion picture sequences. These and further advantageous techniques and features will be apparent to the skilled reader in the following description of specific embodiments of the invention.

Embodiments of the invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Figure 1 illustrates the problem of motion vector estimation in the MPEG coding system;

Figure 2 shows in block form a parallel DSP system suitable for implementing methods embodying the invention;

Figure 3 illustrates the operation of the system of Figure 2 in accordance with the invention;

Figure 4 illustrates the allocation of ordered subsets of pixel blocks between four DSPs in a method embodying the invention;

Figure 5 illustrates the generation of a set of candidate vectors in accordance with a first

search strategy;

Figure 6 shows three alternative patterns for selection of starting vectors; and

Figure 7 illustrates the generation of a set of candidate vectors from a starting vector in accordance with a second search strategy.

The embodiment to be described encodes digitised video images for storage and reproduction in accordance with the MPEG decoder specification, cited above. Within this specification, there are many layers of compression encoding and much flexibility for improved encoding strategies to provide greater image quality and/or compression, and the present description is concerned chiefly with improvements in the method of motion vector estimation, which will not be defined in the standard. For full details of the MPEG coding system, reference can be made to the MPEG specification or the article by Le Gall, both cited above.

Figure 1 illustrates certain steps in the motion vector estimation for a sequence of five frames I0,B1,B2,P3,B4,P5 of pixel data which form part of a motion picture sequence to be encoded. MPEG defines three types of frames, with different degrees of motion compensated compression. The sequence (any sequence) begins with an I-typeframe (intrapicture) I0, which is coded with reference only to itself, and relies on no information from neighbouring frames. The sequence continues with 'batches' of predicted frames, including B-type (bidirectional predicted) frames and P-type frames (predicted).

The batch size M can vary typically from one to four frames, and each batch comprises (M-1) B frames followed by a P frame. Thus in the example sequence of Figure 1, frames B1,B2,P3 form a batch of M=3 frames, while frames B4 and P5 form a second batch with M=2. It will be apparent that a batch with M=1 comprises a single P-type frame. The arrows at the foot of Figure 1 indicate that each P frame is to be 'forward predicted' from the preceding I or P frame, while each B frame is to be both forward and backward predicted from the I and/or P frames on either side.

The principle of motion compensated prediction will now be described briefly, with reference to the forward prediction of frame P3. Each frame to be predicted is divided into an array of blocks of 16x16 pixels, represented (on an enlarged scale) by the block A in frame P3. Having encoded the starting or reference frame I0 for the sequence, the block A is not directly encoded, but rather a motion vector VA is encoded which predicts that the block A is the same as a 16x16 pixel block A' in the reference frame I0. Data compression is thereby achieved by temporal redundancy reduction. Since the two blocks A and A' will not generally be exactly the same, the difference (prediction error)

is then also coded with spatial redundancy reduction by the discrete cosine transform technique (DCT), working on smaller blocks of 8x8 pixels.

Thus the first problem faced by the MPEG coder is that of motion vector estimation for each block in the current frame P3: how to identify the block A' in the reference frame (IO) which best matches the current block A, and hence identify the best motion vector VA? MPEG allows motion vectors to take a limited range in x (horizontal pixel dimension) and y (vertical) respectively where in the latest specification these ranges Rx and Ry are both ± 64 pixels. This creates a limited forward search area FSA, shown in Figure 1, in which to locate the best matching block A', but a full search even of this area would require the evaluation of many thousand candidate blocks. For each candidate block, $16 \times 16 = 256$ pixel comparisons must be made, and the squared pixel differences accumulated (for example) to arrive at a mean squared difference value by which to measure the matching of the candidate block. While a full search may be possible with dedicated hardware, it is presently desired as mentioned in the introduction above to implement motion compensated image coding in software, using readily available processing hardware.

Outside the field of MPEG encoding, de Haan and Huijgen, cited above, have developed motion vector estimation methods which measure matching for relatively few candidate blocks, based on the observations (i) that there is generally a correlation between motion vectors of neighbouring blocks and (ii) that subjective image quality can best be maintained if the motion vector field is smooth, rather than accurate. In other words, there is no great loss in subjective picture quality if the estimated motion vectors are constrained to vary only a little from block to block within a frame. Thus, within the large search area FSA, only a few candidate vectors are used for matching, and these candidate vectors are based on starting vectors estimated previously for blocks neighbouring the current block, spatially and/or temporally.

While the techniques of de Haan and Huijgen can be adapted to a software implementation, which operates many times faster than one with a full-search, this is still several hundred times real time, even with a high performance processor. The present inventors have looked to systems of parallel digital signal processor chips (DSPs), such as the ANDROX ICS-400 image processor board, available from ANDROX in Boston, Massachusetts, and illustrated in block form in Figure 2.

The system of Figure 2 comprises a circuit board housing four processors, referenced DSP0 to DSP3, which can be controlled by a host computer system via a VME bus interface unit 200. The

board can be housed for example inside a Sun Microsystems or similar workstation-type computer, which provides a convenient and familiar development environment. The VME interface is connected to a high speed bus 202, through which flows all information processed by the DSP system. A bus controller 204 is provided to manage such operations. A memory 206 is connected to the bus for the storage of the video data frames being encoded.

Each processor DSP0-3 is connected to the bus 202 by a pair of cache memories CM0 and CM1. Each cache memory CM0 or CM1 can in practice hold the current block of 16×16 pixel values, together with the pixel values of the search area FSA and other data necessary to estimate a motion vector for the block, although to this end the range (Rx,Ry) of candidate vectors is limited in the present embodiment to ($\pm 48, \pm 48$). In operation, one cache of the pair is connected to the DSP at a time, while the other cache can be unloaded and reloaded with data for processing via the bus 202. In this way, a constant supply of fresh data is available to keep the respective DSP working full-time. Each DSP further has a pair of program memories PM0 and PM1. Memory PM0 is used to store the program for the DSP, while memory PM1 is used to store temporary values and data which is common to several blocks.

Figure 3 shows briefly in block form how the task of MPEG coding has been partitioned in the present embodiment. Original pixel data for a batch of M frames (plus reference frame) is received at IMIN, and a first module INTV performs the search for motion vectors to integer precision for the whole batch (for example frames B1,B2,P3), using exclusively the original pixel data. Only this first module INTV need be described in any detail for an understanding of the present invention, but the remaining modules will be described briefly as follows. Four modules HLFV/MDEC/FDCT, Q/RLC, RLD/DEQ and IDCT/REC form a main loop of the encoding process. The module HLFV/MDEC/FDCT receives the integer motion vectors and calculates predicted blocks and the resulting prediction errors, making mode decisions and performing the forward DCT compression in accordance with MPEG published recommendations. In this process, the motion vectors are 'fine tuned' to half-pixel precision.

Module Q/RLC quantises and run-length codes the DCT information and passes these to a module BITS/REG which generates the final MPEG bit stream. The module BITS/REG also provides regulatory feedback to the quantiser module to maintain the output bit rate at the desired value (typically 1.2 Mbits per second). Returning to the main loop, modules RLD/DEQ and IDCT/REC reconstruct the pixel data actually encoded, which is fed back to

allow module HLFV/MDEC/FDCT to make correct mode decisions, locate the best half pixel vectors and so forth. Finally a module DIS allows display of the original or reconstructed pictures.

In the implementation of these modules, the parallel system of four DSPs can only provide a four-fold increase in processing power over a single DSP if all four DSPs are busy all the time. To this end, the ANDROX system is interrupt driven so that each DSP operates independently, but this creates problems for the implementation of the known methods of vector estimation in module INTV, because of the need to predict starting vectors from the vectors of neighbouring blocks. If neighbouring blocks are assigned to different DSPs, and particularly since the estimation may take a shorter time for some blocks than others, there will be times when DSP2, say, requires a starting vector which has yet to be calculated by DSP3. Because of this, much of the potential for parallel processing would remain unexploited.

Figure 4 shows how the blocks of a frame can be partitioned into ordered subsets, as part of a solution to the above problem. In this embodiment, the blocks are divided into their horizontal rows, and all the blocks of a row are processed, in left to right order, by a single DSP. In the particular embodiment illustrated, every fourth row is processed by the same DSP, with the rows relevant to DSP0 being highlighted in the Figure. In connection with the alternation of usage of the CM0 and CM1 caches, alternate blocks are shaded differently to illustrate which block is processed in which cache. Processing the blocks in horizontal rows is preferred for most natural image sequences, where horizontal motion predominates. There is in principle no reason why the blocks should not be processed in vertical rows (columns), diagonal rows or other sequences.

It will now be explained how the module INTV estimates integer motion vectors, firstly for the blocks of the first batch of frames B1,B2,P3, represented by block A in Figure 1, and then for the blocks of subsequent batches, represented by block B in frame P5. Two alternative vector search strategies are embodied in the implementation to be described, depending on whether the reference frame (I- or P-type) being used for the current frame has itself been motion predicted. (It may be noted here that I frames are often encoded in a sequence to provide an 'entry points' into the sequence, even though these frames can be motion predicted; for such frames it is in fact advantageous to find motion vectors to enable use of the second search strategy, even though these vectors will not be encoded in the MPEG bit stream). Both strategies rely on the choice of a good starting vector for the search, from which is generated a set

of candidate vectors, pointing to a corresponding set of candidate pixel blocks within the reference frame.

Referring now to the vector search for block A, in accordance with the scheme of Figure 4, the row of blocks containing block A are all processed in left to right order by DSP3. Thus it is guaranteed that the block LB(A) immediately to the left of the current block A on the current frame P3 will have been processed already by DSP3, and an estimated motion vector will thus be available which predicts the block LB(A) from some block in the reference frame IO. This vector can be used as one starting vector on which to base a search for the motion vector that will predict block A from a block A' in frame IO. On the assumption that no motion vectors have been found for the reference frame IO, nor yet the frames B1 and B2, a limited area-search strategy is used for the blocks of frame P3, as illustrated by Figure 5.

Figure 5 shows a spiral search pattern which, on the integer pixel grid, extends ± 15 pixels in the horizontal direction (Δx) and ± 8 pixels vertically (Δy). This horizontal bias is provided in recognition of the fact that horizontal motion predominates in a typical motion picture sequence, as mentioned above. The program memory PM1 of the relevant DSP contains an ordered list of offset pairs ($\Delta x, \Delta y$) which define a set of candidate vectors 'o' relative to a starting vector, indicated by '+' at offset (0,0). The sequence of candidate vectors 'o' spirals generally outward from the starting vector at '+' to fill the search area up to $(\Delta x, \Delta y) = (\pm 15, \pm 8)$. In this way, even with a single starting vector, 527 candidate vectors 'o' are defined. Each candidate vector in turn identifies a candidate block in the reference frame IO to be compared with the current block A which it is desired to predict.

For each block in the fourth row of blocks in the frame P3 to be predicted, the DSP must therefore identify a starting vector. For the first block in a row, the zero vector (0,0) is used as a default starting vector, but thereafter the vector just estimated for the block LB(A) to the left of the current block is used as the starting vector. This brings important benefits since in most cases there will be a high degree of coherence between neighbouring vectors in the vector field. Having identified the starting vector '+', the stored sequence of offsets can be used to generate the set of 527 candidate vectors 'o' in the spiral sequence.

For each candidate vector 'o', the current block A in frame P3 is compared with a candidate block in reference frame IO whose spatial position (x,y) is given by the candidate vector relative to the spatial position (x,y) of the origin of the current block. The comparison is performed pixel by pixel in principle for all 256 pixels in the block, accumulating to a

mean squared difference (or other value measuring the degree of similarity between the blocks), until the best matching candidate block is identified. It may be noted, however, that a candidate block can in principle be rejected as soon as the accumulated difference exceeds the difference found for the best candidate block tried so far, resulting in a considerable saving of time. Since checking the accumulated squared difference against the best so far will generally occupy at least one cycle of the DSP's time, to make such a check after every pixel would slow down the entire matching process. A good compromise has been found in an embodiment where the accumulated squared difference is checked against the best so far only at the end of each row of 16 pixels in the block.

While only a few per cent of the possible motion vector area FSA (Figure 1) is explored by this spiral search pattern, it is found that the estimated vectors converge upon the true motion vector within a few blocks of the picture's edge. It may be noted that inaccuracies of the motion vectors do not directly reduce the quality of the corresponding blocks in the image when decoded and displayed. Rather, inaccurate motion compensation merely reduces the degree of compression that can be achieved by the next stage in the MPEG coding process, namely spatial redundancy reduction by DCT.

Although far fewer than the number of candidate blocks involved in a full search of the area FSA (Figure 1), 527 candidate blocks per block of the current frame is still a heavy computational task. The ordering of the candidates into a spiral brings potential savings, however, on the assumption that motion vectors are generally coherent from one block to the next. Because of this coherence, the starting vector chosen will, more often than not, be close to the desired motion vector for the current block. The ordering of candidates according to distance from the starting vector means that the best matching candidate block will tend to be found sooner rather than later in the sequence of 527 candidates.

By incorporating in the DSP programming a set of test conditions designed to detect the best candidate as soon as it appears, the search can be terminated, on average, after relatively few candidate blocks have been compared to the current block. One such condition is the setting of a threshold value of the accumulated mean squared difference. Below this threshold (which allows, among other things, for noise in the pixel data), the blocks are judged a 'perfect' match, and the spiral search is terminated. Thus, for example, the generation and evaluation of candidate vectors 'o' might end with the discovery of a 'perfectly' matching candidate block corresponding to the starting vector

plus the offset $(\Delta x, \Delta y) = (3, -2)$, as illustrated in Figure 5. Up to this point, only 48 candidate blocks have been compared.

Around the location of the best integer candidate vector 'o', represented at $(\Delta x, \Delta y) = (3, -2)$ in Figure 5, nine crosses 'x' indicate the nine candidate vectors that will be explored to half-pixel precision by the module HLFV/MDEC/FDCT, described above with reference to Figure 3.

Clearly, the order of candidates can be modified simply by modifying the list of offsets $(\Delta x, \Delta y)$ stored in the DSP program memory PM1. The range of offsets Δx and Δy can be expanded or decreased as desired, and circular or elliptical spirals or other sequences can be implemented, offering some improvement over the square and truncated spiral of Figure 5.

Figures 6 and 7 illustrate the second vector search strategy, to be applied, by way of example to blocks in frame P5, represented by the block B. The strategy to be used for frame P5 compares far fewer candidate blocks even than the limited area search strategy described above with reference to Figure 5. This second strategy relies heavily on the choice of a few starting vectors, at least one of which should be a good prediction of the motion vector for the current block. Coupled with the rule that the vector field is better smooth than accurate, even these few candidates are found to allow a 'good' yet extremely fast motion vector estimate.

Turning to Figure 1, the block LB(B) immediately to the left of the current block B is again known to have been processed already, because of the ordered allocation of the whole row of pixel blocks to a single DSP. The vector just estimated for block LB(B) can therefore be used as one starting vector. Unfortunately, no other neighbouring blocks in the current frame P5 are known to have been processed at this time. The strategy implemented therefore looks to the neighbouring blocks which are known to have been processed, namely neighbouring blocks in neighbouring frames, such as the current reference frame, the previous P frame P3.

For the representative block B, two blocks UR-(B) and DR(B) are shaded in frame P3, these being spatially diagonally up-and-right and down-and-right neighbours of the block B respectively, as well as temporal neighbours of the current block B (frame P5). The two neighbouring blocks provide a further two starting vectors for use in estimating the motion vector of block B.

It may be noted at this point that such starting vectors may need to be scaled versions of the vectors estimated for the neighbouring blocks. For example the vector VA predicting block A in frame P3, say, from block A' in frame IO represents motion of some image feature over a time period of

three frame intervals ($M=3$), whereas a vector predicting block B in frame P5 from a block B' (not shown) in frame P3 represents motion over two frame intervals only. Thus the vector estimated previously for block UR(B) must be scaled by a factor of 2/3 to obtain a good starting vector for the vector estimation of block B, and so forth. (If backward prediction vectors, found for blocks in a B frame, are being estimated or used as starting vectors, negative scaling factors will also be required).

Figure 6A shows the pattern formed by the chosen neighbouring blocks around the current block, with different hatching to distinguish blocks which are spatial neighbours in the same frame ($t=T$) and blocks which are both spatial and temporal neighbours ($t \neq T$). The chosen neighbour blocks generally form a triangle around the current block, with a slight horizontal bias to the distribution. The horizontal bias is in recognition of the fact that in most natural motion picture sequences, horizontal motion predominates. Choosing neighbour blocks to the left and right of the current block helps to ensure that at least one of the starting vectors will be suitable when an object moving into the current block comes from the left or the right.

Figure 6B shows an alternative pattern of neighbouring blocks to yield three starting vectors, and Figure 6C shows a pattern yielding four starting vectors, without horizontal bias. Always in Figure 6 the block LB to the left of the current block is in the current frame ($t=T$). Of course other patterns can be readily envisaged by those skilled in the art, each having its own merits and demerits, depending on the motion picture content, and the pattern can be changed by changing the contents of the program memories PM0, PM1 for each DSP. The pattern of Figure 6A has been found most satisfactory in the present embodiment, for a range of picture sequences. Of course, yet different patterns will be required where the allocation of blocks to the DSPs is different to that of Figure 4. The block rows might run vertically or even diagonally across the picture, or the subsets might be convoluted and/or interleaved for whatever reason; the principles disclosed herein can still be applied.

Having obtained three starting vectors for the current block A, the DSP in the present embodiment evaluates the block matching for a group of up to 21 candidate vectors based upon these three starting vectors.

Figure 7 shows how seven candidate vectors, marked by circles 'o', are defined by a predetermined pattern of offsets ($\Delta x, \Delta y$) relative to the starting vector, marked '+'. The pattern of seven candidates 'o' includes the starting vector '+' itself, and again has a horizontal bias in recognition of the predominance of horizontal motion in the human

environment. By applying this pattern of seven offsets to each of the three starting vectors, a set of 21 candidate vectors is obtained. Before any time-consuming block matching calculations are performed, however, the candidate set is checked to eliminate duplicates and candidates outside the legal range.

Once the best matching candidate vector 'o' has been found, again a final comparison is performed in module HLFV/MDEC/FDCT (Fig. 3) with nine half-pixel candidate vectors, marked as 'x' around the rightmost candidate vector 'o' in Figure 7. This leads to a final encoded motion vector for the current block, estimated to half-pixel precision and derived using reconstructed pixel data.

It will be apparent to the reader that evaluating at most 21 candidate vectors instead of the thousands required for a full search of the area FSA (Figure 1) achieves a significant time saving, but with the disadvantage that the vector estimated for the current block may not in fact be the most accurate one. The subjective quality of images decoded after encoding by this method is good, however, because of the above mentioned observation by de Haan and Huijger that the motion vector filed should be above all smoothly-varying, rather than strictly accurate. Moreover, when object features moving in an image are many pixel blocks in size, the slowly varying vector field will tend to converge upon the most accurate vector within a few blocks after entering the object.

Figure 1 also shows how, on the assumption that motion vectors have been estimated already for the interpolated frame B4, as well as the reference frame P3, a further four possible starting vectors are obtainable from by scaling the forward and backward vectors estimated for the UR and DR neighbouring blocks in frame B4, shaded with dotted lines in Figure 1. Thus a total of up to six starting vectors are available in addition to the vector estimated for the preceding block LB(B). A further four or eight starting vectors could be available for the last frame in a batch of three or four frames. Rather than evaluate a set of seven candidate vectors for every one of these starting vectors, the present embodiment in fact first evaluates only the starting vectors themselves, before forming candidate sets around the starting vectors which give the best matching candidate blocks. Thus, for example, candidate sets according to the pattern of Figure 7 may be formed on the basis of vectors estimated for the block LB(B), the block UR(B) from frame P3 and the backward prediction vector for the DR block in frame B4, all scaled as necessary to account for the different time intervals spanned.

In practice, the vector search for each batch of M frames is performed in the order indicated by

circled numbers by the curved arrows at the foot of Figure 1. By adopting this order, and with appropriate scaling, forward and backward prediction vectors of neighbouring blocks in the interpolated frames B1 and B2 are all available as candidate starting vectors when estimating motion vectors for frame P3. Similarly, the forward vectors for frame B1 can be scaled and used as starting vectors in the forward vector search for frame B2.

The remainder of the MPEG encoding process involves variation of many parameters such as the batch size M and quantisation scaling, in order to achieve a desired compromise between picture quality and data compression, as described in the MPEG specification and the article by Le Gall. These aspects are outside the scope and purpose of the present description, but it has been found that the complete MPEG coding of motion picture sequences including the motion estimation methods described above can be achieved at only 90 times real time using the four DSPs of the ANDROX system, mounted in a Sun workstation. Thus a few minutes of full motion video can be encoded for storage on CD-ROM or similar on an overnight basis, using readily available hardware in a small, compact system.

A further modification of the above strategies can be made when the encoding scheme results in some vector values being more economical to encode than others. In the particular examples of the MPEG encoding format, motion vectors are encoded in the MPEG bit stream by variable length codes, such that shorter vectors require fewer bits. Thus, when two candidate vectors have almost equal matching quality, the modified strategy will favour the vector which requires fewer bits to encode. This may necessitate a little more error information being encoded for a given block, but there will often be a net saving in encoded data which allows the image to be encoded with an overall improvement in quality.

A particular feature of the MPEG variable length codes for motion vectors is that zero displacements Δx and Δy can be encoded in only one bit each. Moreover, at least in a p-type frame a block having zero motion ($\Delta x = \Delta y = 0$) can be coded with no motion information at all. This is signalled by the appropriate variable length code for "macroblock type" in the MPEG bit stream. Thus there is a great economy of data in encoding blocks which have zero motion from one frame to the next. Thus an overall improvement can be obtained when the evaluation of candidate vectors is modified to favour the zero vector, when the matching quality obtained is only slightly below that of the best candidate.

Of course, in some sequences, the zero vector will rarely be among the best candidates. It may be

noted however, that certain types of motion picture sequence, for example animated cartoons, contain a large number of blocks with substantially zero motion. Moreover, large areas of these blocks comprise uniform areas of colour, for example blue sky. Conventional motion compensation techniques will nevertheless estimate and encode motion vectors for these blocks, if only because they contain noise information which "moves" from frame to frame. For encoding these sequences, it has been found that very significant quality improvements are afforded by use of the invention in its third aspect: if the noise level is low, the zero vector will be just one of a large number of candidate vectors for which the matching quality is the same within a relatively narrow threshold.

From reading the present disclosure, many modifications will be apparent to persons skilled in the art, in addition to those mentioned above. Such modifications may involve other features which are already known in the fields of picture encoding, motion estimation and digital signal processing and which may be used instead of or in addition to features already described herein. Although claims have been formulated in this application to particular combinations of features, it should be understood that the scope of the disclosure of the present application also includes any novel feature or any novel combination of features disclosed herein either explicitly or implicitly or any generalisation thereof, whether or not it relates to the same invention as presently claimed in any claim and whether or not it mitigates any or all of the same technical problems as does the present invention. The applicants hereby give notice that new claims may be formulated to such features and/or combinations of such features during the prosecution of the present application or of any further application derived therefrom.

Claims

1. A method of estimating motion vectors for blocks of pixels in a frame of a digitised motion picture sequence, the method comprising, for each given block of pixels performing a vector search based around at least one starting vector derived from a previously estimated vector for a pixel block which is a neighbour in space and/or time of the given pixel block, characterised in that the estimation of motion vectors for each frame is performed in parallel for at least two ordered subsets of the pixel blocks of the frame, the division and ordering of pixel blocks into the subsets and the selection of the neighbouring pixel block(s) being performed such that, where a neighbouring block whose estimated vector is to be used to

- predict that of the given block is selected from the same frame as the given block, the said neighbouring block is one of the same subset as the given block and is ordered in the subset so as to precede the given block.
2. A method as claimed in Claim 1 wherein each subset comprises a row of pixel blocks, the processor estimating motion vectors for the pixel blocks of the row in a predetermined sequence, the motion vector for each given block (disregarding the first block in each row) being predicted by the processor from the motion vector estimated for a block which precedes the given block in the said sequence.
3. A method as claimed in Claim 1 or 2 wherein for each given pixel block at least one further prediction of the motion vector is based on a motion vector estimated for a pixel block in a neighbouring frame in the motion picture sequence.
4. A method as claimed in Claim 3 wherein for each given pixel block in a row (disregarding the first block in each row) predictions are based upon the previously estimated motion vectors of the said preceding block in the same row and a plurality of blocks in the said neighbouring frame.
5. A method as claimed in any of Claims 1 to 4 wherein the predictions for each given pixel block (disregarding the first block in each subset) are based on motion vectors estimated previously for:
- the said preceding block in the same subset as the given block; and
 - two blocks in a neighbouring frame occupying spatial positions relative to the given block so as to define with the said preceding block in the same subset as the given block a triangular distribution generally surrounding the given block.
6. A method as claimed in any preceding claim wherein on the basis of the at least one starting vector, matching between the given block and a block in a previous frame is measured for an array of from 5 to 10 candidate vectors distributed around the starting vector.
7. A method as claimed in any of Claims 1 to 5 wherein on the basis of the at least one starting vector matching between the given block and a block in a previous frame is measured sequentially for an ordered series of candidate vectors surrounding the starting vector and
- 5 with generally increasing distance from the starting vector.
8. A method as claimed in Claim 7 wherein the evaluation of candidate vectors is terminated if a vector is found giving a matching better than a predetermined threshold.
- 10 9. A method of estimating a motion vector for a given block of pixels in a frame of a digitised motion picture sequence by performing an evaluation of a plurality of candidate vectors based around a starting vector, characterised in that the candidate vectors surround the starting vector and are evaluated in an ordered series of generally increasing distance from the starting vector.
- 20 10. A method as claimed in Claim 9 wherein the evaluation of candidate vectors is terminated if a vector is found giving a matching better than a predetermined threshold.
- 25 11. A method as claimed in any of Claims 1 to 10 wherein the selection of each estimated vector depends not only on the measured prediction quality (matching) of each candidate vector, but also on the relative economy with which the candidate vector can be encoded in a resultant data stream encoding the motion picture sequence.
- 35 12. A method as claimed in Claim 10 wherein the zero vector is most economically encoded and wherein a candidate vector which is zero will be favoured in the selection of each estimated motion vectors.
- 40 13. A method of estimating a motion vector for a given block of pixels in a frame of a digitised motion picture sequence, which estimated vector is to be encoded in a data stream encoding the motion picture sequence, the method comprising:
- generating a set of candidate vectors for evaluation;
 - evaluating the quality of prediction provided by each of the candidate vectors; and
 - comparing the results of said evaluations to select one of the candidate vectors as the estimated vector for encoding in a stream of data encoding the motion picture sequence,
- 45 wherein the amount of data required to encode the estimated motion vector is dependent on the value of the estimated motion vector and wherein the selection of the estimated vector
- 50
- 55

from among the candidate vectors depends not only on the quality of prediction provided by each candidate, but also on the amount of data required to encode each candidate in the data stream, such that the candidate vector selected is not the one providing the highest quality of prediction, if it is determined that another one of the candidate vectors can be coded in less data while providing a prediction quality which is within a pre-determined threshold of the highest prediction quality.

5

10

14. A method as claimed in any preceding claim wherein the motion vectors and prediction error information are encoded in accordance with the MPEG standard as defined by the International Standards Organisation.

15

15. An apparatus for encoding a digitised motion picture sequence comprising:

20

- means for dividing at least one given frame of the sequence into a plurality of blocks of pixels;
- means for estimating by the method of any preceding claim a motion vector predicting each block of pixels in the given frame from a block in a neighbouring frame and encoding said motion vector; and
- means for generating and encoding prediction error information to be used with the motion vectors in reconstructing the encoded image.

25

30

16. A method of storing a digitised motion picture sequence on a record carrier, comprising:

35

- using an apparatus as claimed in Claim 15 to encode the motion picture sequence; and
- storing the resultant encoded motion vector information and prediction error information on the record carrier.

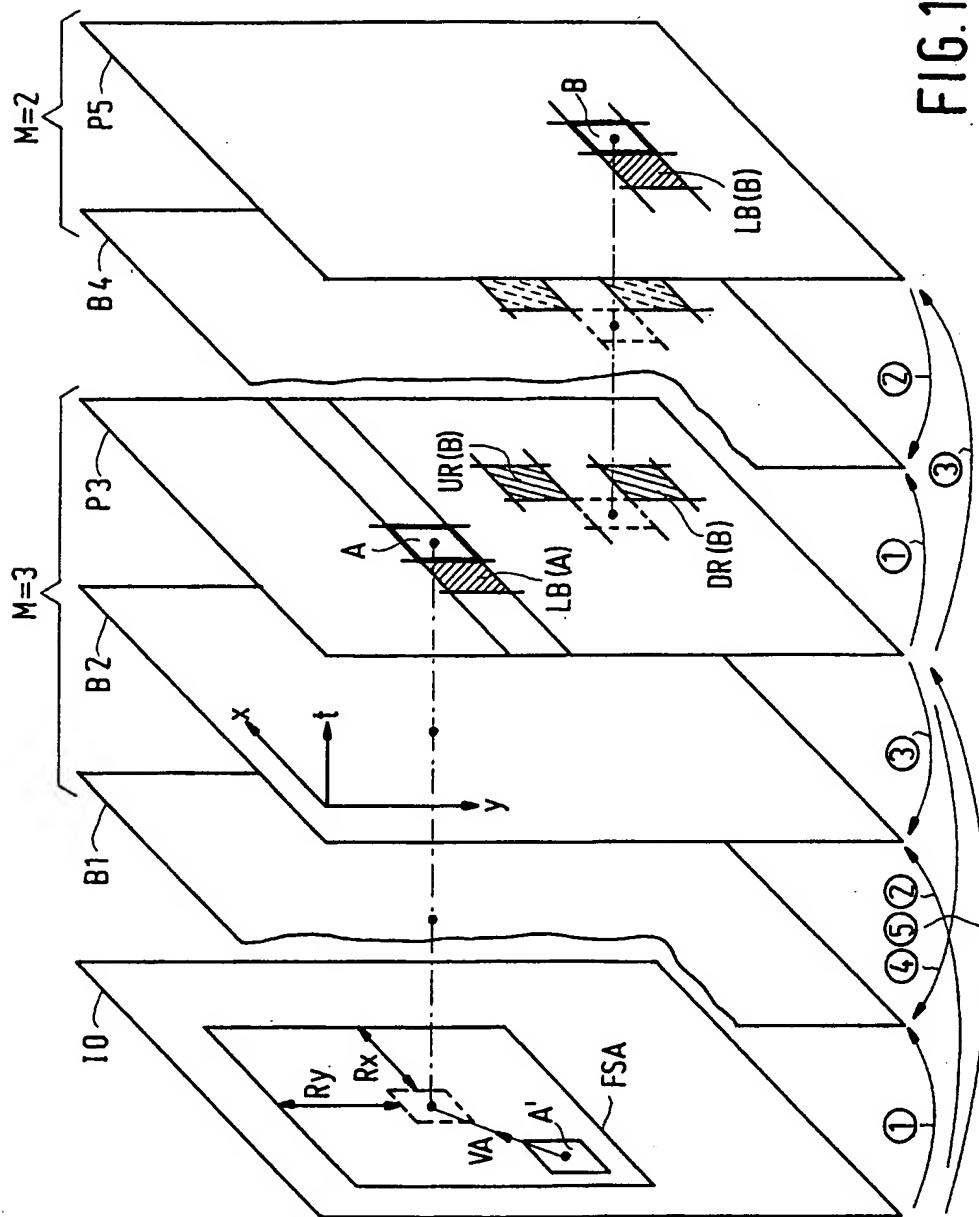
40

17. A record carrier wherein a motion picture sequence has been stored in accordance with a method as claimed in Claim 16.

45

50

55



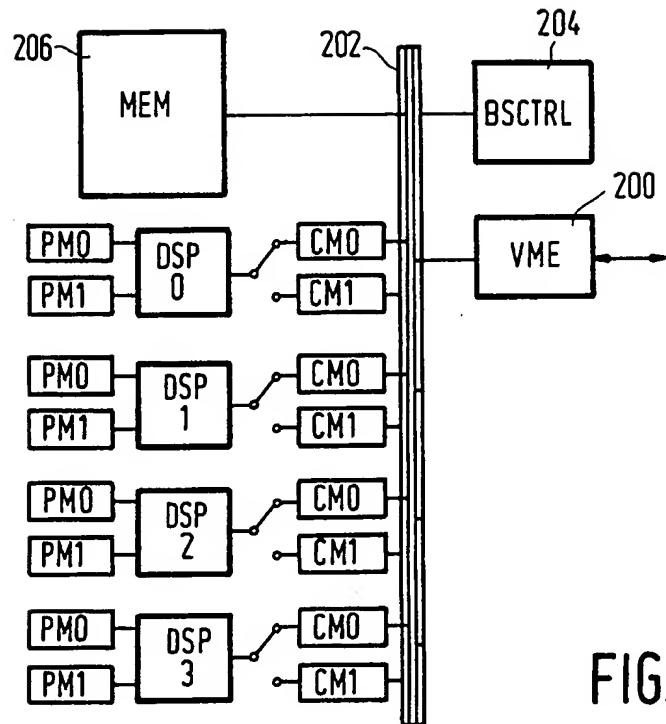


FIG.2

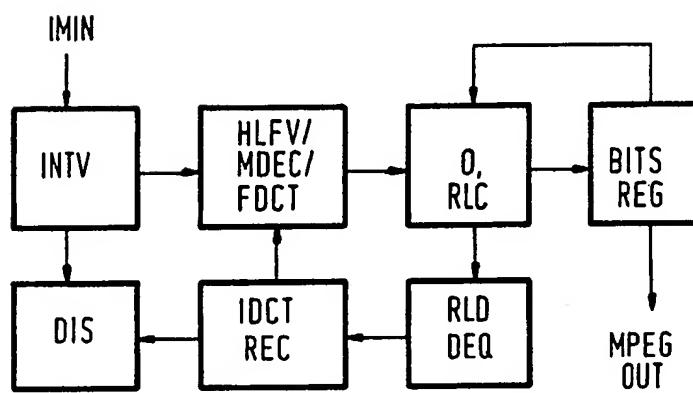


FIG.3

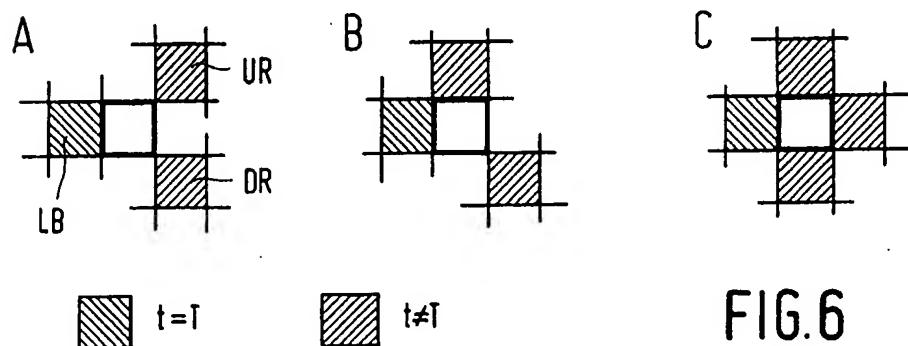


FIG.6

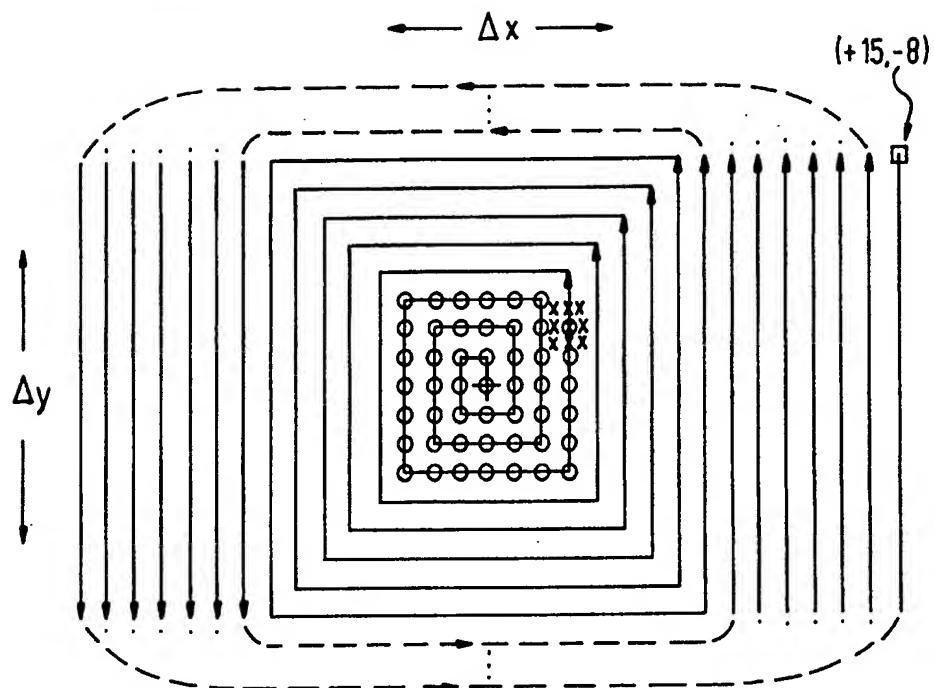
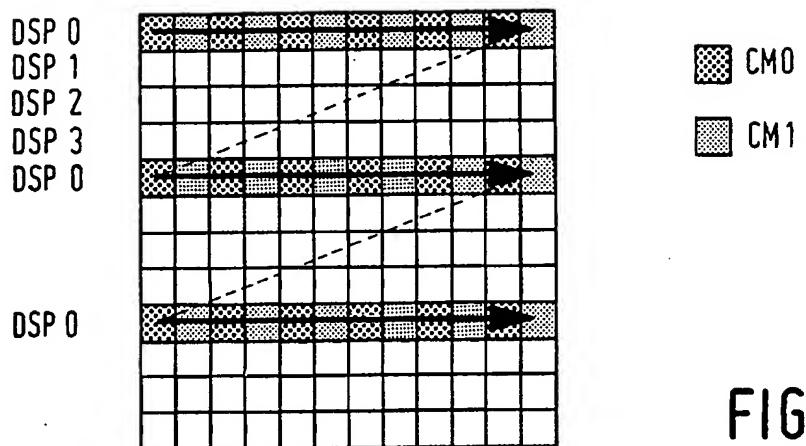


FIG.5

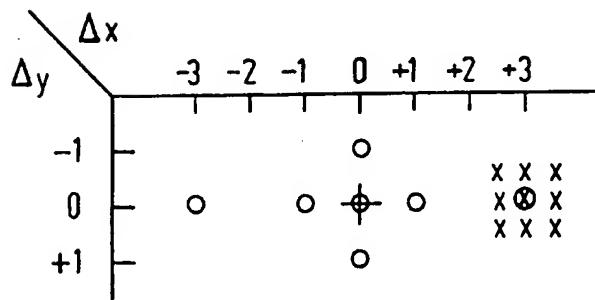


FIG.7